Homework 2

Problem 4: Two-layer perceptron

For the last problem I wrote the following programm in MATLAB:

```
trainingSet = csvread('training_set.csv');
trainingTargets = trainingSet(:, 3);
trainingSet(:, 3) = [];
validationSet = csvread('validation_set.csv');
validationTargets = validationSet(:, 3);
validationSet(:, 3) = [];
trainingSetSize = size(trainingSet,1);
pVal = size(validationSet,1);
learningRate = 0.02;
epochs = 10^2;
iterations = 10^5;
M1 = 8;
M2 = 4;

threshold1 = zeros(M1, 1);
threshold2 = zeros(M2, 1);
threshold3 = 0;

w1 = 0.4*rand(M1, 2)-0.2;
w2 = 0.4*rand(M2, M1)-0.2;
w3 = 0.4*rand(M2, 1)-0.2;

b1 = zeros(M1, 1);
b2 = zeros(M2, 1);
b3 = 0;

V1 = zeros(M1, 1);
V2 = zeros(M2, 1);
O = zeros(trainingSetSize,1);

for ep = 1:epochs

  for i = 1:iterations
    mu = randi(trainingSetSize);
    wSum1 = 0;
    wSum2 = 0;
    wSum3 = 0;

  % Forward Propagation using training set

    for j = 1:M1
      wSum1 = 0;
      for k = 1:2
        wSum1 = wSum1 + w1(j,k)*trainingSet(mu,k);
    end
    b1(j) = wSum1 - threshold1(j);
    V1 = tanh(b1);
  end

  for l = 1:M2
    wSum2 = 0;
    for k = 1:M1
      wSum2 = wSum2 + w2(l,k)*V1(k,1);
    end
    b2(l) = wSum2 - threshold2(l);
    V2 = tanh(b2);
  end

  for m = 1:M2
    wSum3 = wSum3 + w3(m,1)*V2(m,1);
  end

  b3 = wSum3 - threshold3;
  O(mu) = tanh(b3);

  % Backpropagation

  delta3 = (trainingTargets(mu)-O(mu))*(1-O(mu)^2);
  delta2 = delta3*w3.*(1-V2.^2);
  delta1 = transpose(transpose(delta2)*w2).*(1-V1.^2);

  wUpdate3 = learningRate*delta3*V2;
  thresholdUpdate3 = -learningRate*delta3;
  w3 = w3 + wUpdate3;
  threshold3 = threshold3 + thresholdUpdate3;

  wUpdate2 = learningRate*delta2.*transpose(V1);
  thresholdUpdate2 = -learningRate*delta2;
  w2 = w2 + wUpdate2;
  threshold2 = threshold2 + thresholdUpdate2;

  wUpdate1 = learningRate*delta1.*trainingSet(mu,:);
  thresholdUpdate1 = -learningRate*delta1;
  w1 = w1 + wUpdate1;
  threshold1 = threshold1 + thresholdUpdate1;

end % num of iterations

% Forward propagation using validation set

for mu = 1:pVal
  wSum1 = 0;
  wSum2 = 0;
  wSum3 = 0;

  for j = 1:M1
```

```
      wSum1 = 0;
      for k = 1:2
        wSum1 = wSum1 + w1(j,k)*validationSet(mu,k);
      end
      b1(j) = wSum1 - threshold1(j);
      V1 = tanh(b1);
    end

    for l = 1:M2
      wSum2 = 0;
      for k = 1:M1
        wSum2 = wSum2 + w2(l,k)*V1(k,1);
      end
      b2(l) = wSum2 - threshold2(l);
      V2 = tanh(b2);
    end

    for m = 1:M2
      wSum3 = wSum3 + w3(m,1)*V2(m,1);
    end

    b3 = wSum3 - threshold3;
    O(mu) = tanh(b3);

  end

  % Error Calculation

  sum = 0;
  for t = 1:pVal
    sum = sum + abs((sign(O(t))-validationTargets(t)));
  end
  error = sum/(2*pVal);
  fprintf('Validation Error:');
  disp(error);

  if error<0.12
    csvwrite('w1.csv', w1);
    csvwrite('w2.csv', w2);
    csvwrite('w3.csv', w3);
    csvwrite('t1.csv', threshold1);
    csvwrite('t2.csv', threshold2);
    csvwrite('t3.csv', threshold3);
    break
  end

end % num of epochs
```